VLSI Implementation of Discrete Wavelet Transform

by
Vivek T. D.

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
March, 2001

VLSI Implementation of Discrete Wavelet Transform

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

Vivek T.D.

to the

Department of Electrical Engineering

Indian Institute of Technology, Kanpur

March 2001

Certificate

This is to certify that the work contained in the thesis titled "VLSI Implementation of Discrete Wavelet Transform," by Vivek T.D. (Roll No. 9910488) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

29th March 2001

Dr. B. Mazhari
Department of Electrical Engineering,
Indian Institute of Technology, Kanpur.

Abstract

The discrete wavelet transform (DWT) is one of the important tools for signal processing. Real time computation of the DWT often requires dedicated VLSI implementations. In this thesis, the design of a VLSI circuit that computes DWT of a sequence of input samples using 6-tap filters for 3 stages of wavelet decomposition is described. The circuit was designed starting with a behavioral model and passing through register-transfer logic, gate, and physical design levels of the VLSI design flow. A hierarchical design methodology was adopted for RTL level design and logic synthesis, and a standard cell approach was followed for physical design. The VLSI DWT circuit has an area of 10 mm^2 and can operate at a maximum speed of 40 MHz under typical operating conditions. At this speed, the VLSI circuit can process colour frames of video signals in real time.

Acknowledgements

At the outset, I wish to thank my thesis supervisor, Dr B. Mazhari, for his guidance throughout the course of the thesis work. His suggestions on various aspects of this work have indeed improved its quality. It has been a very good learning experience for me working under his guidance.

I am thankful, also to Dr. Aloke Dutta, Dr. K.S. Venkatesh, and Dr. Sumana Gupta, whose courses were very useful for this work.

I take this opportunity to thank Mrs. Nandini Mudgil for extending all possible assistance at the VLSI / EDA lab during the course of the project.

Words of appreciation are also due to my friends, association with whom provided memorable moments. The fruitful discussions I had with them not only enhanced my knowledge but also contributed to my thesis.

I wish to express my deepest sense of gratitude to my parents and sister for being a source of support and encouragement always.

Vivek T.D.

DEDICATED TO MY

PARENTS

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The wavelet transform has emerged as an important signal processing tool. Real world signals have statistical properties that often vary over time. Representation of such signals in terms of functions localized in time called "wavelets," can provide better insight into their properties than the Fourier representation. One of the important aspects of wavelet transform is the relation of continuous time functions with discrete-time filter banks leading to the Discrete Wavelet Transform (DWT) [1]. This concept has given rise to several applications like multiresolution signal processing in computer vision and subband coding for speech and image compression [2]

Many applications require that DWT be computed in real time. Dedicated Very Large Scale Integration (VLSI) systems are often necessary to meet such performances. Although VLSI technology enables realization of fast and compact circuits, the design of such circuits is complex. In order to cope up with the complexity, it has been found useful to break down the design process into different levels of abstractions [3]. A typical VLSI design flow is shown in Fig. 1.1. At the behavioral level, the system is captured in the most abstract form. The system is described in terms of relationship between its inputs and outputs. At the register-transfer logic (RTL) level, the architecture is defined. Information about the resources within the system and their utilization is thus, added. The functional aspects of the system within a set of specifications can be verified at this stage.

The rest of the steps of design flow are technology dependent. At the gate level of abstraction, the system is realized with logic gates and other digital blocks pertaining to a specific technology. Estimates of various parameters like area, path delay etc., can be obtained after gate level design of the system. The next level of design flow involves planning of system implementation on a silicon wafer, placement of transistor level cells of digital blocks synthesized at the gate level, and routing of signals, all according to technology specific design rules. These are collectively referred to as physical design. Accurate information regarding parasitic capacitances, signal delays and area is obtained after this step. The circuit can be tested for all the specifications it is required to meet after the physical design of the system.

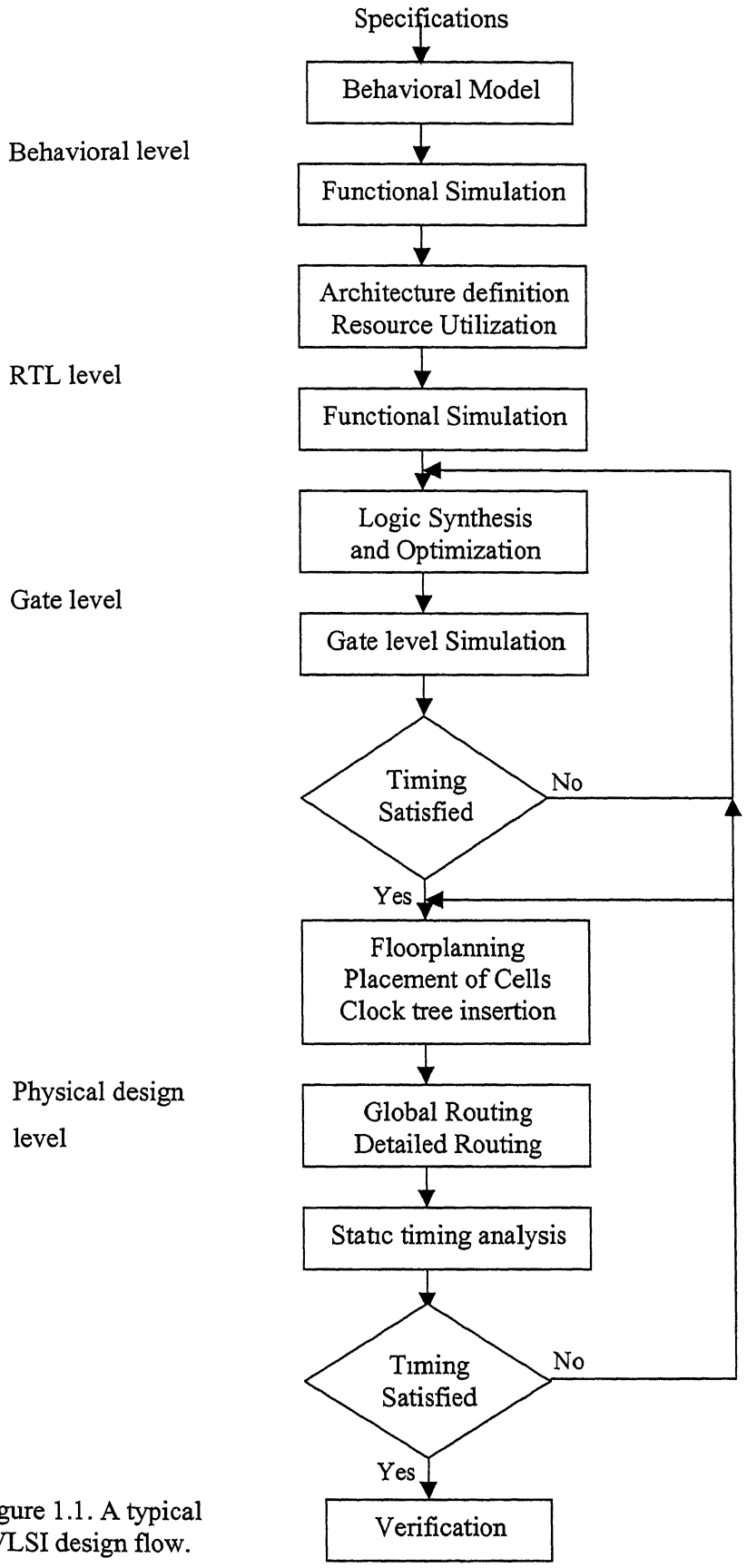

Figure 1.1. A typical VLSI design flow.

Details such as architecture, technology etc., may need to be added to the system while moving from one level of the VLSI design flow to another. Decisions to this effect need to be taken at possibly all levels of the design flow. Some of the decisions taken may turn out to be incorrect due to the VLSI system not satisfying the specifications or the design not being optimal. In such a case the design process may have to be repeated in part or full with a fresh set of decisions. Hence, the design process is iterative. The knowledge base formed previously can aid in taking such decisions.

The focus of this thesis is to design a VLSI circuit that computes the DWT of a semi-infinite sequence of input data in a 'running' manner, passing through all the levels of VLSI design flow shown in Fig.1.1. Attempts have been made in this work to optimize the circuit with respect to speed. A hierarchical design approach is used for RTL and gate level design while a standard cell approach [3] is followed for physical design. The decisions taken at different levels of abstraction are reviewed for their correctness after results obtained at the physical design level are analyzed. In the next section, a review of previous work is given. The chapter is concluded with a description of organization of the thesis.

1.1 Literature Review

A number of VLSI architectures have been proposed for implementation of the discrete wavelet transform. Most of them are 'bit parallel' in nature, wherein all the bits of an input sample are used for computation in a single clock cycle. Architectures for implementation of the DWT are based on Mallat's pyramid algorithm [1]. Two computation schedules exist for the pyramid algorithm. They are referred to as 'As soon as possible (ASAP)' and 'As late as possible (ALAP)' schedules in this thesis.

VLSI architecture for DWT was first proposed in [5] along with the ASAP schedule. The architecture uses two filter units, one each for high pass and low pass filters, along with a set of register files for the storage of intermediate DWT coefficients. Multiplexers are used to switch the appropriate samples for filtering at their respective clock cycles. The disadvantage of the architectures based on ASAP schedule is that the minimum cycle time of the clock is lower bounded by one convolver time.

Architectures based on ALAP schedule have been proposed in [6,7]. The proposed architectures use a single unit of serially connected registers for storage of intermediate coefficients, but in [7], a single filter unit is used for computation of both high pass and

low pass DWT coefficients as against two in [6]. Apart from bit parallel architectures, a digit serial architecture has also been presented in [6]. With this architecture, specific word sizes alone can be used for representing input samples, filter and DWT coefficients.

In [7], results of an integrated circuit realization of a DWT circuit for 6 tap filter and for 3 levels of wavelet decomposition are given. The circuit, designed with 1.2μm process technology, can function at a maximum clock frequency of 20 MHz, thereby meeting real time requirements for monochrome video processing. It uses 6 multipliers, 6 adders and 26 registers.

In the present work, a modification of the architecture proposed in [6] is used for implementation.

1.2 Organization of the Thesis

The thesis is organized as follows. In Chapter 2, computational aspects of DWT are examined. The pyramid algorithm and the related computation schedules are described. In Chapter 3, behavioral modeling of the system is explained. The VLSI architecture used for DWT circuit implementation is also described. The choices made with respect to the resources and their utilization are outlined, reflecting the modifications made to architecture in [6]. The technology dependent steps of the design flow, as applied to the present system are explained in Chapter 4. The results obtained are presented and discussed. The disadvantages of the circuit are also mentioned. In Chapter 5, possible extensions to this work are suggested.

Chapter 2

Computation of the DWT

2.1 Introduction

The discrete wavelet transform represents a finite energy signal in terms of a family of basis functions referred to as wavelets Such functions can be generated by translation and dilation of a 'mother wavelet' corresponding to the family. For example, if $\psi(t)$ is a mother wavelet, a family of basis functions may be obtained from,

$$\psi_{j,n}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t-2^j n}{2^j}\right)_{j,n\in Z} \tag{2.1}$$

The DWT coefficients can be obtained as an inner product between input signal and the basis functions. A fast algorithm is often advantageous from the computational point of view. In the next section, the pyramid algorithm due to Mallat [8] is described briefly. In Section 2.3, the computation schedules for the pyramid algorithm [5,6] are described.

2.2 The Pyramid Algorithm

The pyramid algorithm gives a fast and efficient way to compute the discrete wavelet transform of a signal. It relates the family of basis functions to a tree structured, maximally decimated filterbank [4]. The schematic representation of the algorithm is shown in Fig. 2.1. The discrete wavelet transform recursively decomposes the input signal $S_0(n)$ into successive approximations and details at the next resolution. If the approximation at stage i is $S_i(n)$, then the approximation $S_{i+1}(n)$ and detail $W_{i+1}(n)$ at stage $i+1$ are given by,

$$S_{i+1}(n) = \sum_{l=0}^{L-1} h(l)\, S_i(2n-l) \tag{2.2}$$

$$W_{i+1}(n) = \sum_{l=0}^{L-1} g(l)\, S_i(2n-l) \tag{2.3}$$

As i increases the resolution of the signal decreases. Thus the input signal $S_0(n)$ may be assumed to be at a higher resolution than $S_i(n)$, $i > 0$. The computation of DWT coefficients can be viewed as a multiresolution decomposition of the signal [8]. The coefficients $h(l)$ and $g(l)$ correspond to taps of low pass filter $H(z)$ and high pass filter

$G(z)$ respectively. The length of the filters L, depends on the wavelets used. From (2.2) and (2.3) it can be seen that, at any stage i, the computation of $S_{i+1}(n)$ and $W_{i+1}(n)$ is equivalent to low pass and high pass filtering of $S_i(n)$ followed by subsampling the output by a factor of 2.

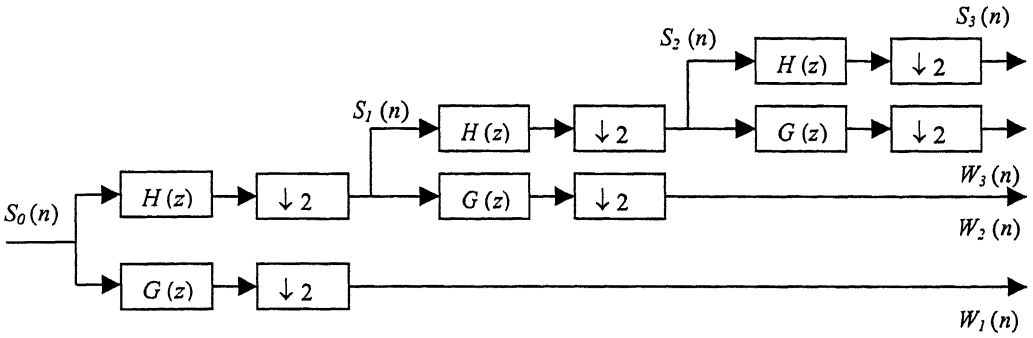

Figure 2 1.The pyramid algorithm for 3 stages of DWT decomposition.

In order to derive the hardware architecture that computes the DWT as shown in Fig. 2 1, the relationships between the inputs, intermediate samples and the outputs must be known. They are represented in a compact manner by (2.2) and (2.3). Equations (2.4a)-(2.4g) give a set of detailed computations for obtaining low pass DWT coefficients with a 6-tap filter and for three stages of decomposition. The high pass DWT coefficients are obtained by merely replacing S_i by W_i on the L.H.S and $h(l)$ by $g(l)$ on the R.H.S of (2.4a)-(2.4g).

Stage 1 DWT Coefficients.

$$S_1(0) = h(0)S_0(0) + h(1)S_0(-1) + h(2)S_0(-2) + h(3)S_0(-3) + h(4)S_0(-4) + h(5)S_0(-5) \quad (2.4a)$$

$$S_1(1) = h(0)S_0(2) + h(1)S_0(1) + h(2)S_0(0) + h(3)S_0(-1) + h(4)S_0(-2) + h(5)S_0(-3) \quad (2.4b)$$

$$S_1(2) = h(0)S_0(4) + h(1)S_0(3) + h(2)S_0(2) + h(3)S_0(1) + h(4)S_0(0) + h(5)S_0(-1) \quad (2.4c)$$

$$S_1(3) = h(0)S_0(6) + h(1)S_0(5) + h(2)S_0(4) + h(3)S_0(3) + h(4)S_0(2) + h(5)S_0(1) \quad (2.4d)$$

Stage 2 DWT Coefficients:

$$S_2(0) = h(0)S_1(0) + h(1)S_1(-1) + h(2)S_1(-2) + h(3)S_1(-3) + h(4)S_1(-4) + h(5)S_1(-5) \quad (2\ 4e)$$

$$S_2(1) = h(0)S_1(2) + h(1)S_1(1) + h(2)S_1(0) + h(3)S_1(-1) + h(4)S_1(-2) + h(5)S_1(-3) \quad (2.4f)$$

Stage 3 DWT Coefficients:

$$S_3(0) = h(0)S_2(0) + h(1)S_2(-1) + h(2)S_2(-2) + h(3)S_2(-3) + h(4)S_2(-4) + h(5)S_2(-5) \quad (2.4g)$$

It is convenient to index the clock cycles with a sequence of non-negative integers starting from 0 to directly interpret the computations of (2.2) and (2.3). Thus, $S_0(0)$ represents the input sample at cycle 0, $S_0(1)$ at cycle 1 and so on. Samples with negative indices are taken to be zero implying that the history of samples is known to be zero. Since the clock under consideration here is to provide a reference for input samples $S_0(n)$, the indices of $S_0(n)$ alone match the clock cycle numbers. It is assumed that the DWT coefficients are fully computed in one clock cycle here.

It can be seen that for a running implementation of the DWT, $S_1(n)$ must be computed in cycles 0, 2, 4 and 6 only. The coefficients $S_2(0)$ cannot be computed before cycle 1 if a single set of filter units are to be used, as $S_1(0)$ required for computing $S_2(0)$ is available only after clock cycle 0 [Equations (2.4a,e)]. Similarly $S_2(1)$ cannot be scheduled for computation before cycle 5 due to its dependency on $S_1(2)$, which in turn depends on $S_0(4)$. The DWT coefficient of the third stage, $S_3(0)$ can be computed in or after cycle 3 but excluding the cycles used to compute lower stage coefficients, as a single set of filter units is assumed to be used. The argument leads itself to the computation schedules described in the next section.

2.3 Computation Schedules for the DWT

For a 'running' implementation of the DWT, it is necessary to compute the DWT coefficients at the same rate as that of input samples. The computation schedules take advantage of the fact that, the first stage decomposition filters need only to compute half the number of coefficients as that of input samples due to subsampling by a factor of 2. The second and third stage filters need to compute one-fourth and one-eighth the number of coefficients as that of input samples. Hence, for an input sequence of N samples, there would be N/2, N/4 and N/8 samples at the output of the first, second and third stage filters respectively [9]. This provides a way to time-multiplex the same set of filter units to compute the DWT coefficients of all the stages.

For three stages of DWT decomposition, computation of DWT coefficients is periodic in the cycle index with a period 8. In one period of 8 cycles there are 4 low pass coefficients of first stage, two of second stage and one of third stage. This is true with high pass DWT coefficients also. The computation schedules are shown in Tables 2.1 and 2.2. In either of them, coefficients of the first stage are scheduled for computation in every other clock cycle. The coefficients of second and third stages are scheduled for

computation every four and eight cycles respectively. Each of the DWT coefficients, S_t and W_t, given by (2.2) and (2 3) are completely calculated in a single clock cycle. The approximation coefficients $S_t(n)$ must be available at the filter inputs before $S_{t+1}(n)$ and $W_{t+1}(n)$ may be calculated. It can be seen from Table 2.1 that $S_1(n)$ needs to be computed before the cycle $8k+1$ starts, whereas with schedule in Table 2.2, it is sufficient for $S_1(n)$ to be present at the inputs of the filters before cycle $8k+3$. This aspect of schedule in Table 2.2 allows for pipelining of filter for faster operation.

Table 2.1 The ASAP Schedule.

Cycle	LPF computation	HPF computation
$8k$	$S_1(4k)$	$W_1(4k)$
$8k+1$	$S_2(2k)$	$W_2(2k)$
$8k+2$	$S_1(4k+1)$	$W_1(4k+1)$
$8k+3$	$S_3(k)$	$W_3(k)$
$8k+4$	$S_1(4k+2)$	$W_1(4k+2)$
$8k+5$	$S_2(2k+1)$	$W_2(2k+1)$
$8k+6$	$S_1(4k+3)$	$W_1(4k+3)$
$8k+7$	-	-

Table 2.2. The ALAP Schedule.

Cycle	LPF computation	HPF computation
$8k$	$S_1(4k)$	$W_1(4k)$
$8k+1$	-	-
$8k+2$	$S_1(4k+1)$	$W_1(4k+1)$
$8k+3$	$S_2(2k)$	$W_2(2k)$
$8k+4$	$S_1(4k+2)$	$W_1(4k+2)$
$8k+5$	$S_3(k)$	$W_3(k)$
$8k+6$	$S_1(4k+3)$	$W_1(4k+3)$
$8k+7$	$S_2(2k+1)$	$W_2(2k+1)$

The detail coefficients $W_t(n)$ and the approximation coefficients S_3(n) constitute the actual outputs of the system.

In schedule of Table 2.1, the DWT coefficients of each stage are scheduled for computation at the earliest possible clock cycle. Hence it is named as 'As soon as possible'(ASAP) schedule. Similarly, the DWT coefficients of each stage are scheduled for computation in as late a clock cycle as possible in the schedule of Table 2.2. Hence it is referred to as 'As late as possible' (ALAP) schedule.

Chapter 3

Behavioral Modeling and RTL Design

3.1 Introduction

The design of the system at RTL level involves definition of the hardware architecture It enumerates all the resources in terms of arithmetic and storage units as also the control signals required to implement the desired function. In the next section, the behavioral model of the system, which forms the basis for certain choices for the RTL level design, is described. In Section 3.3, all the subblocks of the architecture are described The chapter is concluded with a block diagram of the architecture. The results of simulation of design at the RTL level are also shown.

3.2 Behavioral Modeling

The specifications of interest at the behavioral level consist of a description of the function to be realized, the external signals required and their functions. These specifications for the system being designed are given below.

1. The VLSI system must compute the DWT of a stream of input data.
2. All data inputs and outputs are to be controlled by a clock signal.
3. High pass and low pass filters may use upto 6 filter coefficients.
4. The coefficients are to be provided externally at start up, one per clock cycle.
5. A system-wide 'reset' signal is required to completely reset the system when necessary.
6. An output signal 'ready' is required to indicate whether filter coefficients or the input signal samples are being input.

The system, once modeled at the behavioral level, is simulated to validate its functionality. Simulation with test input samples also yields information about the dynamic ranges of intermediate and output samples. This is useful in deciding the number of bits required to represent the samples in the circuit such that overflow errors do not occur.

The system to be designed was modeled using VHDL[1] [10] and simulated for a square wave input sequence shown in Fig. 3.1.

[1] VHDL stands for Very high speed integrated circuit Hardware Description Language

The 6-tap Daubechies wavelet filters given by (cf. page 251, [1])

$h(l)$ = [0 332670, 0.806891, 0.459877, -0.135011, -0 085441, 0.035226] (3.1a)

$g(l)$ = [0 035226, 0.085441, -0.135011, -0.459877, 0.806891, -0.332670] (3.1b)

were chosen for simulation. The DWT coefficients at the three stages of decomposition are shown in Figs. 3.2-3.7.

3.2.1 Choice of word lengths

From an analysis of the values assumed by filter coefficients, input samples and DWT coefficients, the word lengths required to represent them can be determined. They are discussed below.

- The filter coefficients corresponding to orthonormal wavelets, (3.1a) and (3.1b) are fractional. Hence they may be represented in 8 bit 2's complement full fractional format. This is denoted as 1.7 format, wherein the binary point is assumed to be present after the most significant (sign) bit in the 8 bit word.
- The input samples $S_0(n)$ are represented by 8 bit quantization values. This is typical for most signals like image or video data.
- The dynamic range of the outputs of filters for 3 levels of decomposition would increase at each stage by a factor equal to Σ_l abs[$h(l)$] for low pass filters and Σ_l abs[$g(l)$] for high pass filters in the worst case. For coefficients given by (3.1a) and (3 1b), these factors are equal to 1.8551. Hence 4 bits may be needed to represent the integer part of the real number. This is satisfied with 11 bits of quantization (4.7 format) for all DWT coefficients.
- The size of input samples will have to be increased by 3 bits by sign extension in order to match the size for both input samples and DWT coefficients

3.2.2 Errors due to truncation and quantization

The values of DWT coefficients with both real values and finite precision are shown in Figs. 3 2-3.7 for the input in Fig. 3.1. The filter coefficients and the DWT coefficients were represented with 8 bits and 11 bits respectively. The 19 bit partial products were used for accumulation and the sum was truncated to 11 bits. The absolute maximum

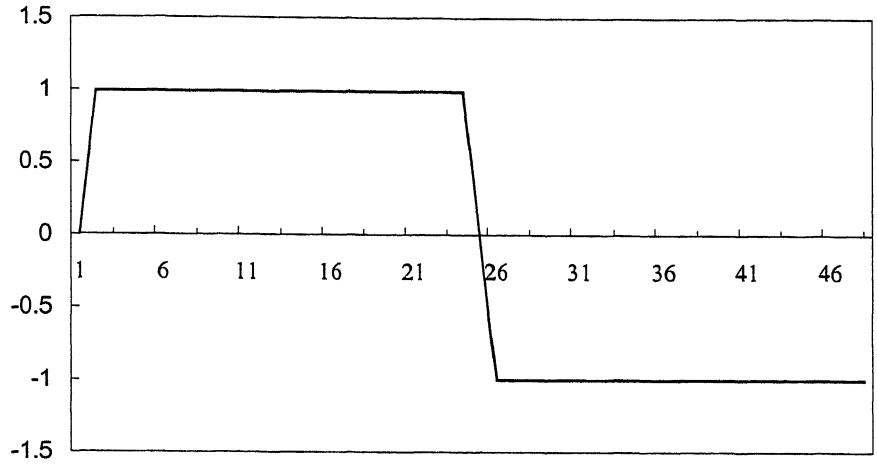

Figure 3.1. Input Samples $S_0(n)$

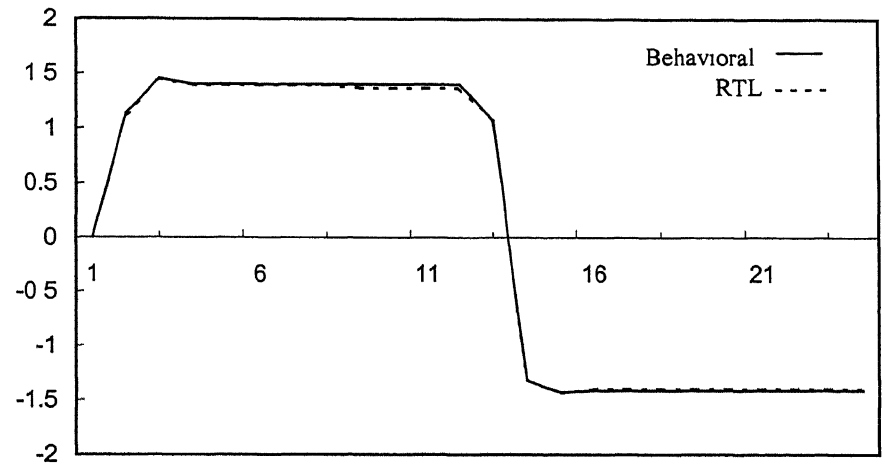

Figure 3.2. Low pass DWT coefficients of first stage $S_1(n)$

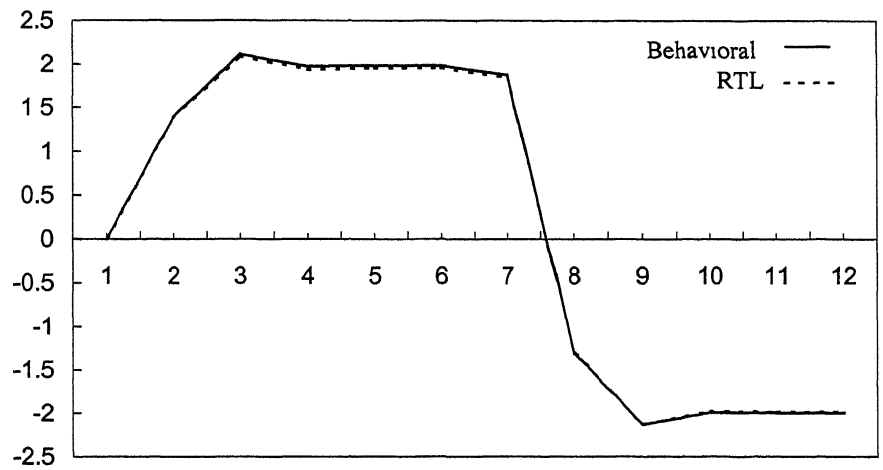

Figure 3.3. Low pass DWT coefficients of second stage $S_2(n)$.

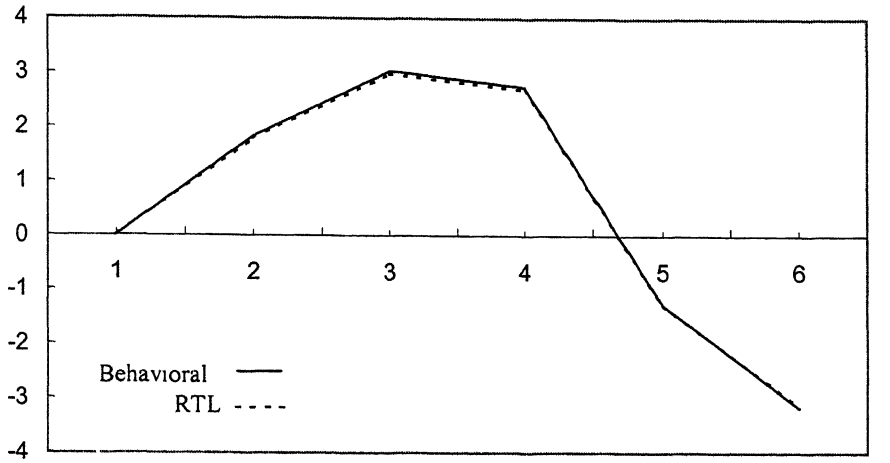

Figure 3.4. Low pass DWT coefficients of third stage $S_3(n)$.

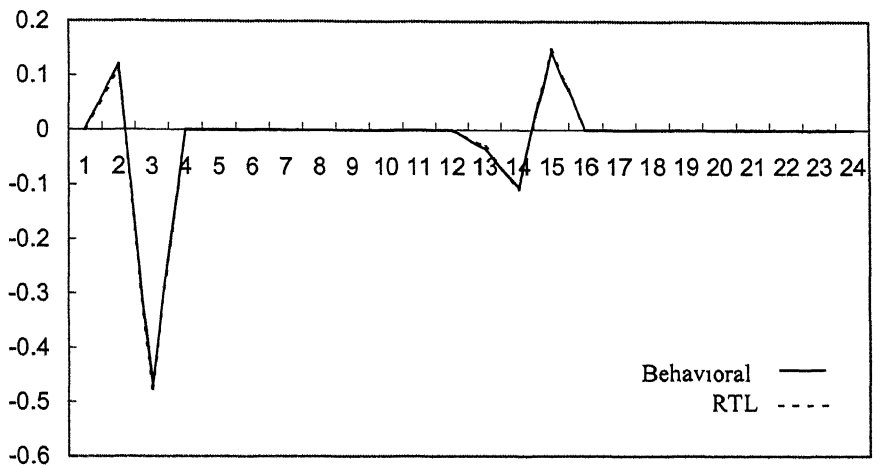

Figure 3.5. High pass DWT coefficients of first stage $W_1(n)$.

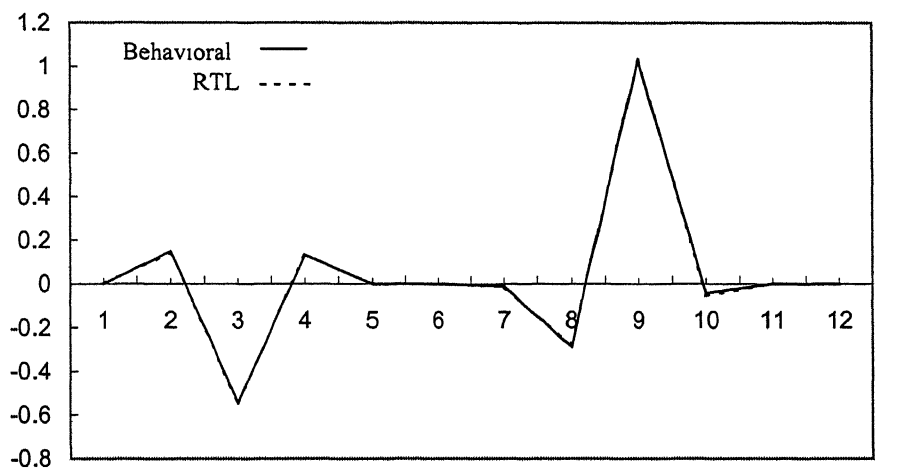

Figure 3.6. High pass DWT coefficients of second stage $W_2(n)$.

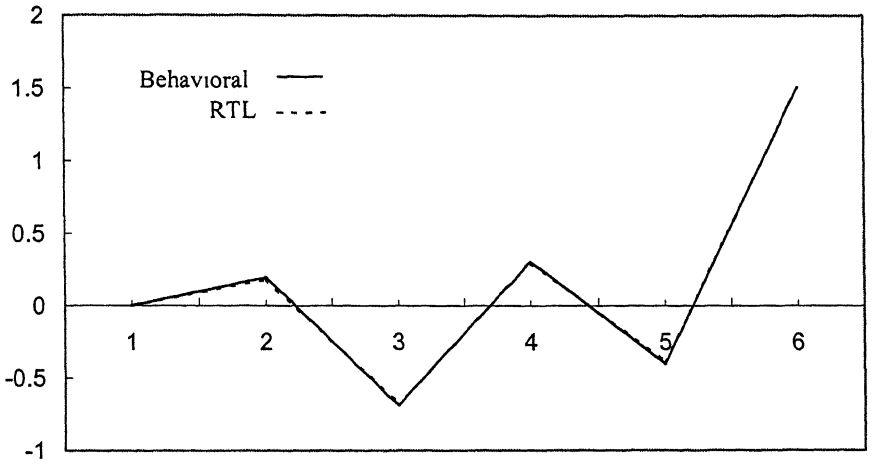

Figure 3.7. High pass DWT coefficients of third stage $W_3(n)$.

errors in the finite precision DWT coefficients due to quantization and truncation for input in Fig. 3.1 are given in Table 3.1.

Table 3.1. Absolute maximum errors due to finite precision effects for input in Fig. 3.1.

Decomposition Stage	LPF outputs	HPF outputs
Stage 1	0.036	0.0103
Stage 2	0 033	0.0121
Stage 3	0.0601	0.0140

3.2.3 Modification for filters with integer coefficients

In order to use the same circuit for filters with integer coefficients, certain modifications must be made while interpreting the fixed-point numbers represented by words. They are discussed with an example below.

Example 3.1

The low pass and high pass filter coefficients corresponding to a bi-orthogonal wavelet are given by (3.2a) and (3.2b).

$$h(l)=[0.125, 0.375, 0.375, 0.125] \quad (3.2a)$$

$$g(l)=[-2.0,2.0] \quad (3.2b)$$

The low pass filter coefficients are fractional and hence no change in the interpretation of low pass filter coefficients is necessary. The coefficients of high pass filters are integers. They must be represented in 3.5 format (instead of 1.7 format) and the outputs of high pass filters need to be represented in 6.5 format (instead of 4.7 format). Thus –2.0 is

represented as 110 00000. The rest of the coefficients for 6-tap filters are assumed to be zeros.

3.3 The VLSI Architecture

It can be seen from (2.2) and (2.3) that, to compute the DWT coefficients, the filter coefficients and the input samples must be convolved and subsampled by a factor of 2. Therefore, a delay chain of length 6 and two filter units consisting of multipliers and adders are required for filters of length 6. In order to store the filter coefficients and the intermediate DWT coefficients, storage elements are needed. A control unit is required to generate internal control signals for proper operation. In the following subsections, the various subblocks of the architecture that lead to RTL design of the complete system are described.

3.3.1 Filter coefficients unit

The coefficients corresponding to low pass and high pass filters are stored in a chain of 12 serially connected registers with parallel outputs. With this structure, it is possible to input 12 coefficients in a predetermined order at a rate of one coefficient per clock cycle. The filter coefficients are latched to the registers at the falling edge of the clock.

3.3.2 Delay unit

For a 6-tap filter, only 5 previous input samples need to be stored along with the present input sample at any particular clock cycle. This is accomplished by using a chain of 6 serially connected registers having parallel outputs. In order to keep the pin count of the integrated circuit minimum, the same set of input lines are used to input both filter coefficients and input samples. A control signal generated by the control unit is used to demultiplex the data on the input lines between the filter coefficients unit and the delay unit. The input samples are also stored in the registers at the falling edge of the clock in this implementation.

3.3.3 Filter units

Two filter units, each containing 6 multipliers and 5 adders are required to perform filtering of the input samples. From (2.2) and (2.3), a common structure shown in Fig. 3.8 can be inferred. If a 'multiply and add' operation is to be performed in one clock

cycle, the minimum cycle time required is T_m+3T_a where T_m and T_a are multiplier and adder times respectively. For long filters, this could be costly in terms of speed of operation.

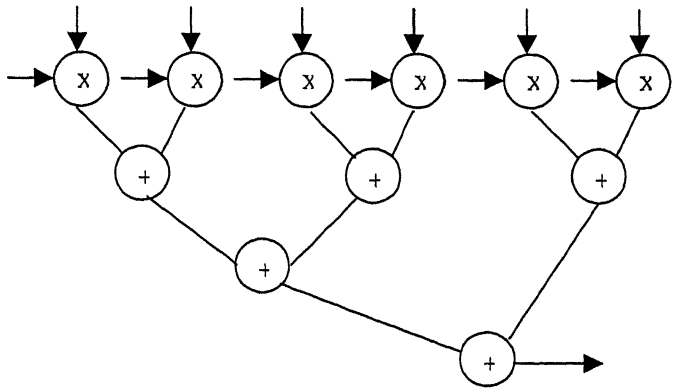

Figure 3.8. A filter unit.

Alternatively, the computation of a DWT coefficient can be partitioned into the following two subcomputations.

1. Computation of partial products $h(l)S_i(2n-k)$ and $g(l)S_i(2n-k)$.
2. Summation of all the partial products.

With a structure shown in Fig. 3.9 for the two filters, the first part can be performed in one clock cycle and the second part in the next clock cycle.

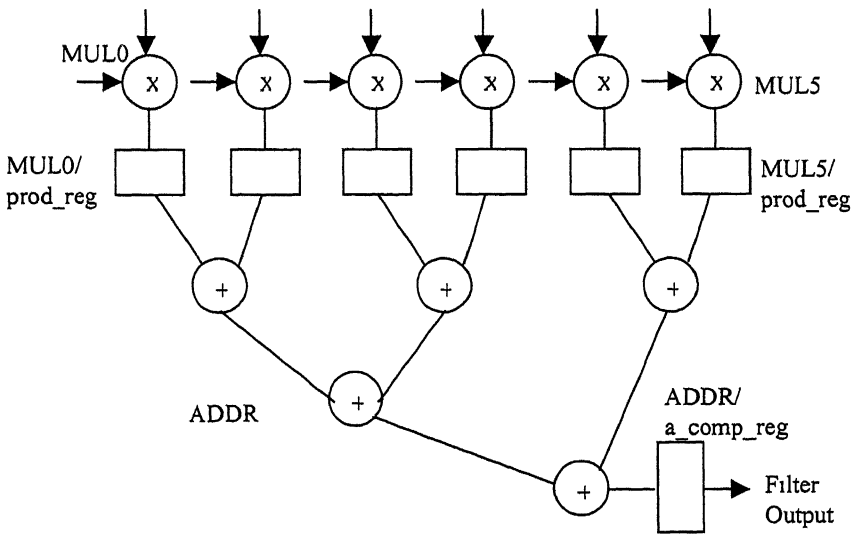

Figure 3.9. A pipelined filter unit.

For convenience, the multipliers and the corresponding registers in Fig. 3.9 are labeled as MUL*i* and MUL*i*/prod_reg respectively, where *i* assumes values from 0 to 5. The chain

of adders and the associated register are labeled as ADDR and ADDR/a_comp_reg respectively. The computation of each of the DWT coefficients $S_i(n)$ and $W_i(n)$ is initiated by computing the partial products at the clock cycles given in Table 2.2 (The ALAP schedule), but is completed *only* in the next clock cycle after summation of all the partial products. The detailed schedule for low pass filter unit is given in Table 3.2. The same schedule also applies to high pass filter unit with $h(i)$ being replaced by $g(i)$. With this structure, the critical path in the filter unit in terms of path delay can be reduced to max(T_m, 3Ta) [11]. The filter unit pipelined by 2 stages has a latency of 2 cycles and a throughput of one DWT coefficient per cycle. At a rate of one input sample per clock cycle, the DWT may be computed in a running manner.

Table 3.2 Computation and output schedule for low pass filter.

Clock Cycle	MUL*i* computation	ADDR computation	Filter Output
$8k$	$h(i)S_0(2n-i)$		
$8k+1$	0	$S_1(4k)$	
$8k+2$	$h(i)S_0(2n-i)$	0	$S_1(4k)$
$8k+3$	$h(i)S_1(2n-i)$	$S_1(4k+1)$	0
$8k+4$	$h(i)S_0(2n-i)$	$S_2(2k)$	$S_1(4k+1)$
$8k+5$	$h(i)S_2(2n-i)$	$S_1(4k+2)$	$S_2(2k)$
$8k+6$	$h(i)S_0(2n-i)$	$S_3(k)$	$S_1(4k+2)$
$8k+7$	$h(i)S_1(2n-i)$	$S_1(4k+3)$	$S_3(k)$
$8k+8$	$h(i)S_0(2n-i)$	$S_2(2k+1)$	$S_1(4k+3)$
$8k+9$	0	$S_1(4k+4)$	$S_2(2k+1)$

3.3.4 DWT coefficients unit

The storage unit for low pass DWT coefficients $S_1(n)$, $S_2(n)$, $S_3(n)$ can be designed once the computation schedule for the filters is known. In this architecture two register files are used for storing previous values of first stage and second stage low pass DWT coefficients $S_1(n)$ and $S_2(n)$ respectively. They are labeled as STG1 and STG2 in Fig. 3.10. The third stage low pass DWT coefficient $S_3(n)$ is temporarily stored in a single register (GREG) before being output in cycle $8k+11$. Since the DWT coefficients of three stages are computed at different rates as described in Section 2.3, three shift signals having appropriate timing relationships are required to clock the two register files and

the single register. Each register file consists of a set of serially connected registers with parallel outputs. All the registers are triggered at the rising edges of respective shift signals. The parallel outputs of register files provide DWT coefficients of a lower stage as inputs for filter units for computation of DWT coefficients of next higher stage. The relationship between the shift signals 'clk1', 'clk2', 'clk3' and the primary clock signal 'clk' is shown in Fig. 3.12.

The register file for storing the first stage DWT coefficients (STG1) consists of 6 serially connected registers, whereas the register file for storing second stage DWT coefficients (STG2) requires 5 serially connected registers. This is because, $S_2(2k)$ is required at one of the inputs of multiplier (MUL0) two cycles after its computation for computing $S_3(k)$. The register at the output of the filter unit, ADDR/a_comp_reg, suffices for storing $S_2(2k)$ for the duration of computation of $S_3(k)$.

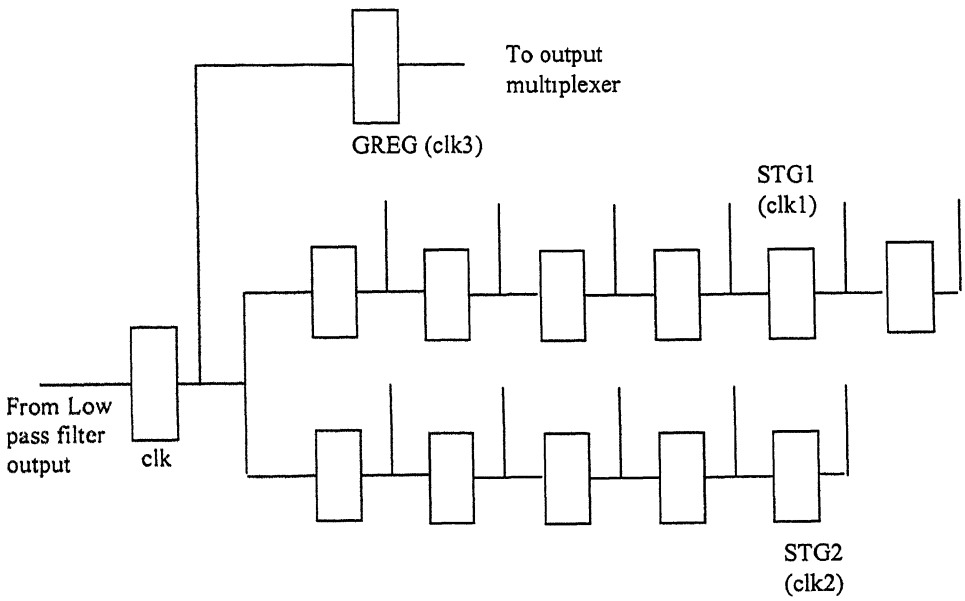

Figure 3.10. The DWT coefficients unit.

3.3.5 Control unit

The control unit is required to generate all the internal signals required for proper functioning of the system. A list of all control signals and their functions are given in Table 3.3. The waveforms of all the control signals are shown in Figs. 3.12 and 3.13. The first task of control unit is to keep track of filter coefficients and generate a signal to select the filter coefficients unit to load it with coefficients. A counter is used for this purpose. At the rising edge of the 13th clock cycle, the signal 'hn_rdy' turns high to

demultiplex the data on input lines to delay unit after 12 filter coefficients are taken in. The signal is defined to change state at the rising edge of the clock in the 13th cycle rather than the falling edge in order to prevent the occurrence of a potential hazard due to signal delays A possible hazard in this case is the latching of input sample by the filter coefficients unit due to a wrong state of 'hn_rdy'. An important outcome of this decision is that it gives rise to half-cycle relationships between registers. The internal signal 'hn_rdy' is also connected to output signal 'ready' to serve as an external indicator for status of 'hn_rdy'.

Table 3.3. Different control signals and their functions.

Control Signal	Function
clk1	Shift signal for registers storing first stage DWT coefficients
clk2	Shift signal for registers storing second stage DWT coefficients
clk3	Shift signal for the register storing third stage DWT coefficients
hn_rdy	Signal to demultiplex data on input lines between delay unit and filter coefficients unit.
s[1:0]	Select lines to switch one of the 4 sets of samples to filter inputs
op_sel	Signal to select $W_i(n)$ or $S_3(n)$ at the output.

From Table 2.2 it can be seen that, computation of DWT for 3 stages of decomposition is periodic in 8 clock cycles. Control signals are necessary to switch a particular set of DWT coefficients of previous stage that must be available at the inputs of filter units for computation of coefficients of the present stage. The signal s[1:0] selects one of the 4 possible samples as inputs to filters. This signal is derived from a mod-8 counter. The counter is enabled after all the 12 filter coefficients are taken in. The truth table in Table 3.4 shows the particular values assumed by the select signal and the set of DWT coefficients selected in the four states.

3.3.6 Output unit

The output unit consists of a multiplexer and an output register. The output select signal 'op_sel' generated by the control unit selects $S_3(n)$ at clock cycle $8k+11$ and $W_i(n)$ at the rest of the cycles for the outputs of the system.

Table 3.4. Select signals at different clock cycles.

Cycle	count	s[1:0]	selection
8k	0	00	S_0
8k+1	1	11	0
8k+2	2	00	S_0
8k+3	3	01	S_1
8k+4	4	00	S_0
8k+5	5	10	S_2
8k+6	6	00	S_0
8k+7	7	01	S_1

3.3.7 Resources within the system and their utilization

The block diagram of the DWT system at the RTL level is shown in Fig. 3.11. It consists of twelve 8 bit × 11 bit multipliers and ten 19 bit adders for computation of the DWT. The storage units for intermediate DWT coefficients and the delay unit consist of thirteen and six 11-bit wide registers respectively. The 12 registers used for pipelining the two filter units are 19 bits wide. The filter coefficients unit consists of 12 8-bit registers connected serially. A single 11-bit register is used to store $S_3(n)$ temporarily.

The filter units are utilized for 87.5% of the time of computations. The register files STG1 and STG2 are clocked 50% and 25% of the time as that of input delay unit clocked by the primary clock signal. Hence the registers in STG1 and STG2 are used 50% and 25% of time respectively. The register GREG is used for only 12.5% of the time. For filters with less than 6 coefficients, the computational resources of filter units are underutilized.

The architecture in Fig. 3.11 differs from the one presented in [6] in that, it uses different shift signals for shifting the contents of registers in the DWT coefficients unit serially as compared to a single clock signal in [6,7]. This avoids a lengthy register file or a minimum length register file with complex interconnections. The generation of shift signals (clk1, clk2, clk3) is not difficult as the periods of signals are multiples of period of the primary clock (clk) signal by factors 2, 4 and 8 respectively, as shown in Fig. 3.12. The architecture was simulated for input shown in Fig. 3.1. The control signals and filter outputs are shown in Figs. 3.12 and 3.13. The DWT coefficients are also plotted in Figs. 3.2-3.7.

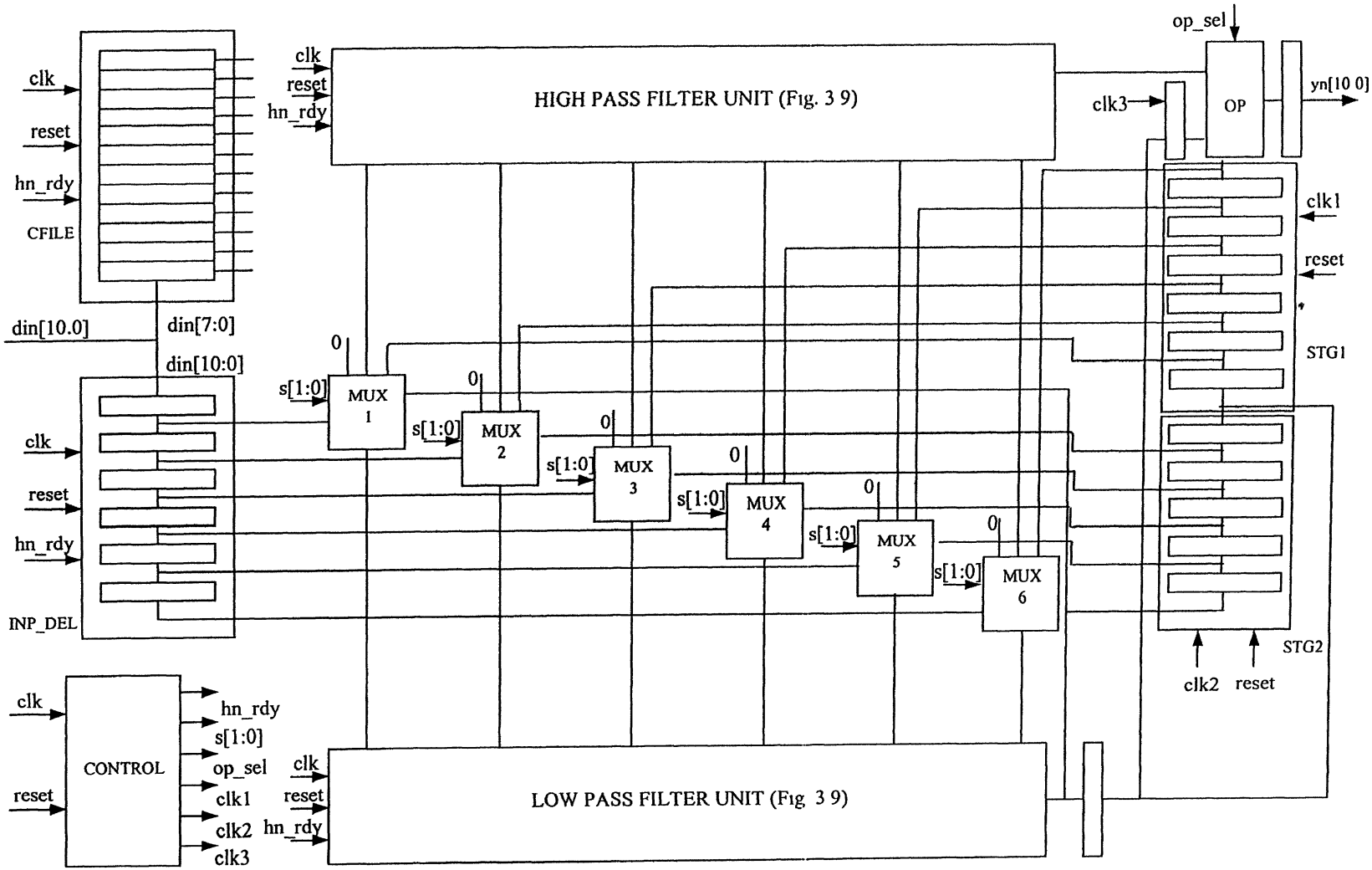

Figure 3.11. The VLSI DWT architecture.

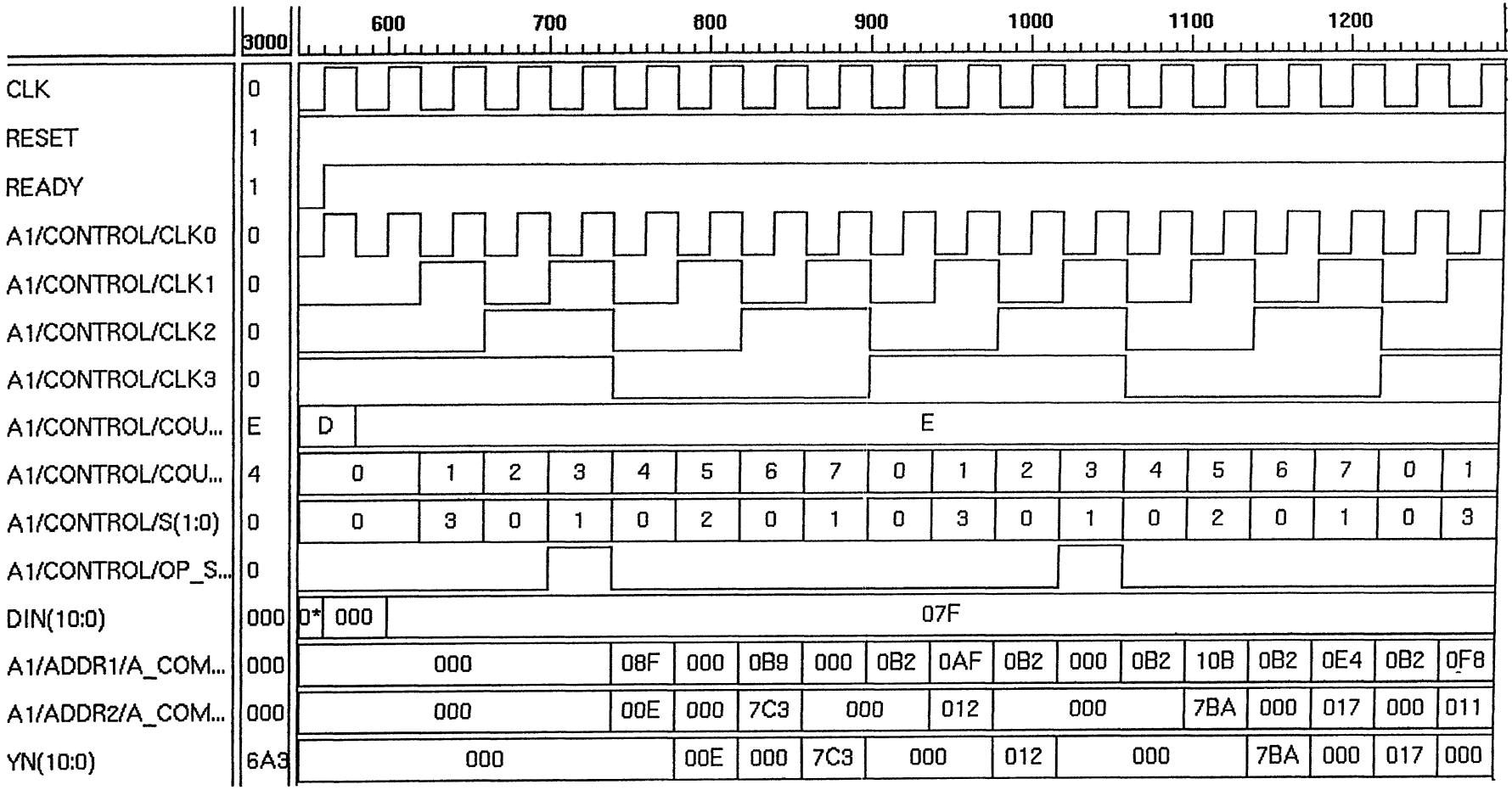

Figure 3.12. Results of simulation of RTL level design

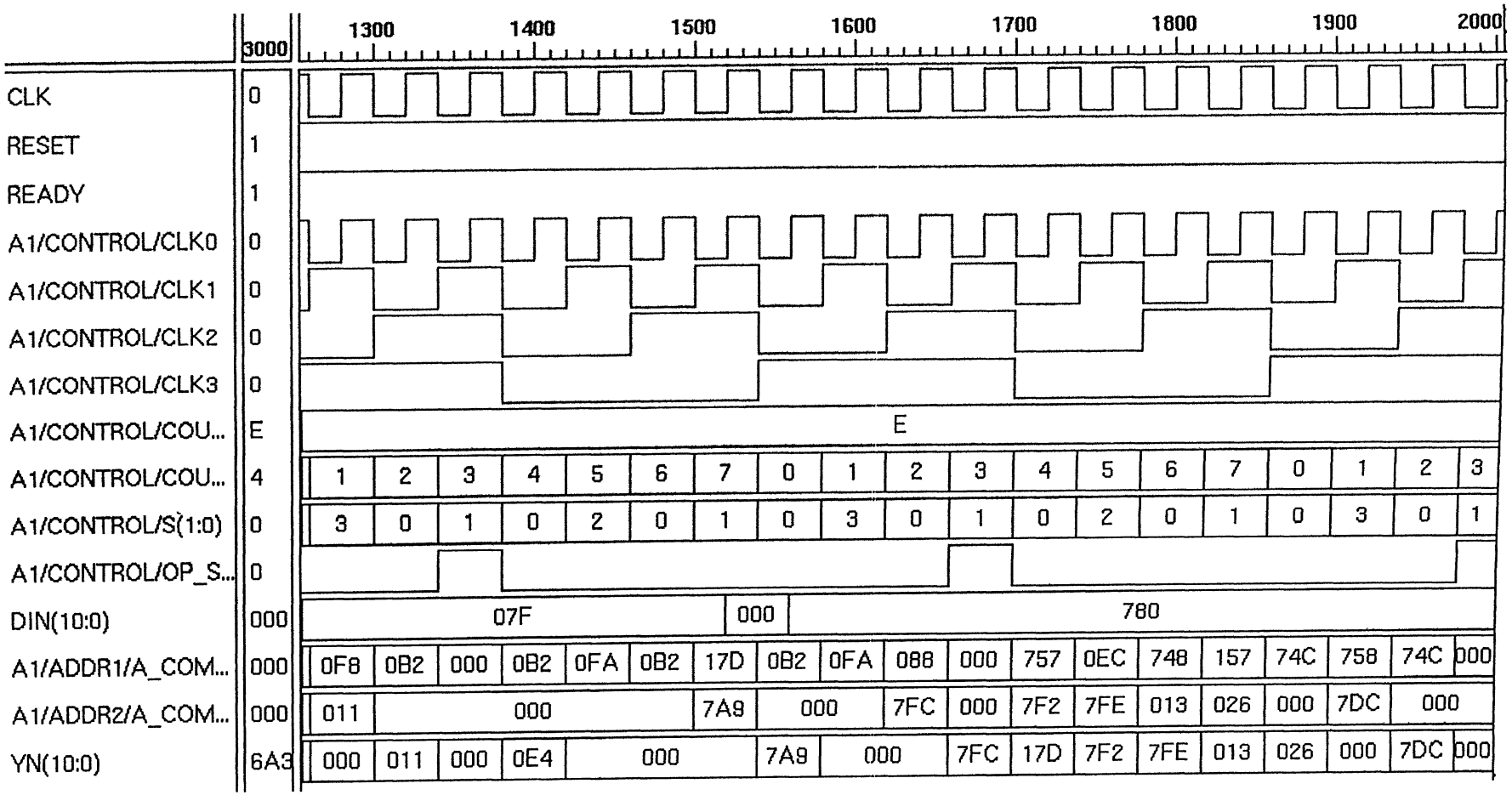

Figure 3.13. Results of simulation of RTL level design

Chapter 4

Implementation and Results

4.1 Introduction

The gate level and the physical design level constitute the technology dependent levels of the VLSI design flow. The gate level is also the first level in the design flow at which technology dependent optimization of the design may be performed. In the next section, the logic synthesis of the DWT architecture and its optimization for speed is described. The results obtained are outlined. In Section 4.3, various steps of physical design as applied to the present VLSI system are explained. The timing aspects of the design are detailed in Section 4.4. The results obtained are compared with those of gate level estimates. In Section 4.5, the results are discussed and some of the decisions taken during the design process are reviewed. The disadvantages of the design are also mentioned.

4.2 Gate level design

Logic synthesis is an important step at the gate level of VLSI design. The aim is to map digital blocks of a particular technology to realize the same function as that of RTL level design, possibly optimizing it with respect to one or more parameters of the design space [14]. The inputs at gate level of design include RTL description of the design and the technology library consisting of different logic gates and other digital blocks. Timing and area constraints are also required for respective optimizations.

4.2.1 The technology library

The technology library used for logic synthesis consists of 87 cells encompassing a variety of logic gates of different input strengths and other digital blocks like flip-flops, multiplexers and adders of 0.6 micron CMOS technology. These cells are used for realizing core logic whereas, a separate library of Input and Output (IO) pad cells are used for synthesizing IO pads for inputs and outputs of the circuit. The cells in the library also provide timing information like gate delays, setup and hold times for sequential elements apart from their functions. These are used for timing optimizations. Different design environments consisting of operating conditions and related derating factors, wire

load models etc., are also provided in the library. This offers the designer the choice to optimize the design for a specific design environment.

4.2.2 Logic synthesis and optimization

The RTL level design was described in a hierarchical manner using synthesizable VHDL. In this approach, various subblocks of the design were described first and then instantiated at a higher level (complete design level). This approach lends itself to hierarchical synthesis of the design followed here. The different subblocks that span the entire design are listed below. They are named as in parenthesis for future reference.

1. Filter coefficients unit (CFILE)
2. Delay unit (INP_DEL)
3. Control unit (CONTROL)
4. 8 bit x 11 bit multiplier 19 bit product register (MUL)
5. 3 stage, 19 bit adder unit with 11 bit sum register (ADDR)
6. 11 bit, 4 x 1 multiplexer (MUX)
7. Output multiplexer with a 11 bit output register (OP)
8. First stage DWT coefficients unit (STG1)
9. Second stage DWT coefficients unit (STG2)
10. 11 bit register (REG)

The logic circuit at the gate level was synthesized from the RTL description of the design using Synopsys Design Compiler [13]. The individual blocks were first synthesized without constraining them. Of the different operating conditions shown in Table 4.1, 'TYPICAL' parameters were used for synthesis.

Table 4.1. Different operating conditions and related parameters.

Operating Conditions	Voltage	Temperature
WORST	2.6 V	100^0C
TYPICAL	3.3 V	25^0C
BEST	4.2 V	20^0C

The subblocks were then instantiated at a higher level and the entire design was synthesized again without applying timing constraints. The objective of such a synthesis was to identify the different cell inputs occurring as loads to outputs of previous cells in the path. The inputs and outputs of cells at the boundary of subblocks may be characterized for loads realistically once they are known. This in turn leads to realistic

characterization of IO pins of cells for transition times and delays, so that they may be taken into account during timing driven optimization.

In the next step, the inputs and outputs of cells were characterized for loads and timing constraints were applied to different paths at the top level with respect to clock. Initially, a specific time period was assumed for the clock and constraints were applied relative to the clock. Based on the timing estimates obtained after synthesis, the cycle time of the clock was reduced and constraints were applied afresh. The synthesis was repeated until when the timing constraints could not be met without any design constraint violation. The design constraints also included a maximum transition time of 0.75 ns and a maximum capacitance of 0.7 pF imposed by the technology library on all the nets. The input and output pads were inserted after core logic circuit was realized. Special buffer cells from an IO pad library were used for this purpose.

4.2.3 Results of logic synthesis

The timing estimates obtained after logic synthesis include delays along different signal paths in the circuit, clock-to-Q delays of flip-flops [14], and data arrival times at points along a path terminated by registers. The path of maximum delay determines the minimum cycle time of the clock and hence the maximum speed of operation. It was found that for clock cycle times less than 22.47 ns there were setup time violations. Therefore the minimum cycle time for clock was estimated to be 22.47 ns. From a gate level timing analysis of the logic circuit, the maximum path shown in Fig. 4.1 was deduced.

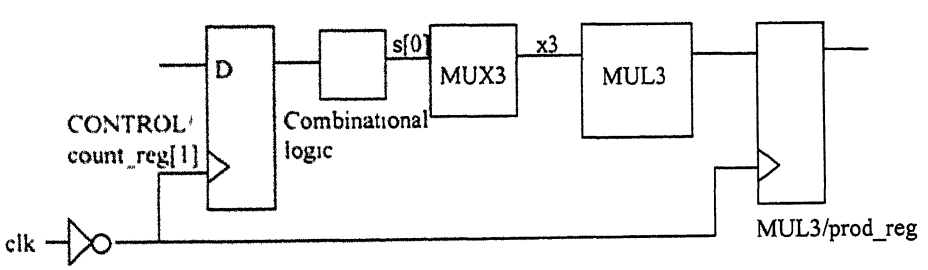

Figure 4.1. The maximum delay path deduced from timing analysis at gate level.

Further, the delays of important digital blocks are given below.

1. Maximum multiplier (MUL3) delay: 12.68 ns
2. Maximum multiplexer (MUX3) delay: 2.33 ns
3. Maximum adder unit (ADDR) delay: 7.61 ns

Estimates of areas of different blocks, synthesized before and after the timing constraints were applied, are given in Table 4.2. It can be seen that area has increased after the design was optimized with respect to time. The increase in the area of multiplier is highest among all the subblocks of the circuit. Since there are 12 multipliers in the DWT circuit, a major part of the core area is occupied by multipliers.

Table 4.2 Area estimates of subblocks before and after timing driven optimization was performed. (All values are in square microns)

Subblock	Area before timing driven optimization	Area after timing driven optimization
REG	13374.7	13582.1
CONTROL	17936.6	25090.5
INP_DEL	80663.0	95696.6
STG1	68428.6	73716.5
STG2	57024.1	59719.6
CFILE	117987.8	120683.5
MUL	167754.5	263761.9
ADDR	120372.5	140382.7
MUX	14411.5	29756.2
OP	16485.1	18869.7

The estimates of areas of the VLSI system at different stages of gate level design are as follows. Area of the design,

1. before timing constraints were applied: 2732901.2 square microns.
2. after timing constraints were applied : 3756430.1 square microns.
3. after IO pads were inserted : 4320190.5 square microns.

4.2.4 Netlist generation

The netlist corresponding to the gate level design was generated using Synopsys Design Compiler in Verilog[1] format. The Verilog netlist was used as an input to layout tool for design of VLSI system at the physical design level.

4.3 Physical Design

The physical design of a VLSI system involves realization of synthesized logic circuit with transistor level cells and wires connecting them. The different inputs at physical design level of design flow are shown in Fig. 4.2. A standard cell library contains geometrical layout information of all the cells used for logic synthesis. It also

[1] Verilog is also a hardware description language like VHDL.

contains information regarding the metal layers used for routing different wires for interconnections between cells.

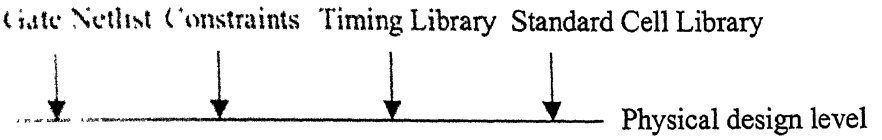

Figure 4.2. Inputs for physical design

Each cell is also characterized by its input, output, power and ground pins. The timing library provides the timing information of all cells. Gate netlist and constraints are related to the design. The netlist was imported into Cadence Silicon Ensemble Place and Route [15] tool along with other libraries and constraints for physical design.

4.3.1 Floorplanning and placement of cells

The purpose of floorplanning is to partition the chip area into core and IO pad regions to place respective cells. In the core region, all the cells corresponding to those in the logic circuit are placed in identical rows whereas the IO pad cells are placed in IO region along with pads for power(VDD) and ground(GND). The floorplan for the DWT circuit is shown in Fig. 4.3.

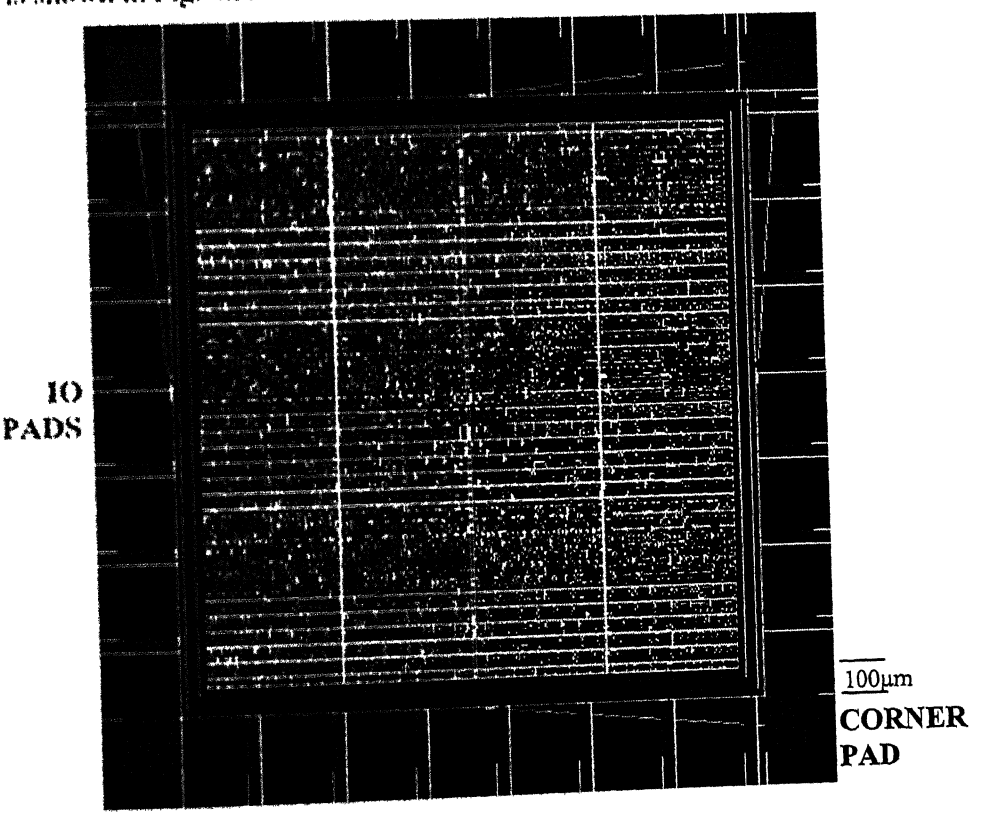

Figure 4.3. Floorplan with core and IO cells placed

The IO pads form the interface between external pins and related internal wires. Prior to placement of cells, the chip area was configured to have an aspect ratio of 1. It implies that the dimensions of both height and width are equal. The 28 IO pads and 4 corner pads for the DWT chip were placed as shown in Fig. 4.3. The core consisting of 5623 cells were placed in 72 rows in the core area. The space of 100 μm between IO pad region and core area was used for power and ground lines, each 40 μm wide.

4.3.2 Clock tree generation

A clock signal provides time reference for sequential elements in the circuit. It is necessary that delays along the clock nets do not affect the timing relationships between different circuit elements like registers of various subblocks [14]. Hence the synthesis of clock tree is an important step. The aim is to achieve acceptable and almost equal delays from the point of application of clock signal to the leaf pins of different sequential elements. In other words, the *clock skew* should be minimum.

In the present circuit, the input clock signal is connected to clock pins of 441 leaf cells. The tree was synthesized using CTGen utility of Silicon Ensemble [15] by constraining the tree for minimum skew iteratively. The least skew obtained was 132 ps with a maximum clock delay of 4.721 ns and a minimum clock delay of 4.589 ns. The maximum transition time along the clock nets was 0.669 ns. In order to achieve this, a clock tree with 7 levels of buffers was synthesized.

4.3.3 Routing of signals

The final step in the design process is to route all the signals. The library offers 4 metal layers (M1-M4) for routing of nets. Among these, M1 and M3 were used for horizontal routing and M2 and M4 for vertical routing. The routing was performed in two phases. In the first phase, routing of global signals like clock, power and ground signals were performed. The other nets in the circuit were routed in the second phase.

4.3.4 Results

The complete VLSI DWT chip is shown in Fig. 4.4. The areas of different regions of the chip and the area occupied by the respective cells are given in Table 4.3. The total area of the chip is 10013427.36 μm^2 (10.01 mm^2). The area occupied by different cells is 5143046.4 μm^2 (5.14 mm^2), thus utilizing 51.36% of chip area.

Table 4.3. Area report (Area values are in square microns)

Chip region	Area of the region	Area of cells	Utilization
IO pad region	3975091.2	631411.2	15.88%
Core region	4352071.6	3765139.2	86.51%
Corner pad region	746496	746496	100%

The dimensions of both width and height of the integrated circuit are 3166.2 μm. The results related to timing aspects are discussed separately in the next section.

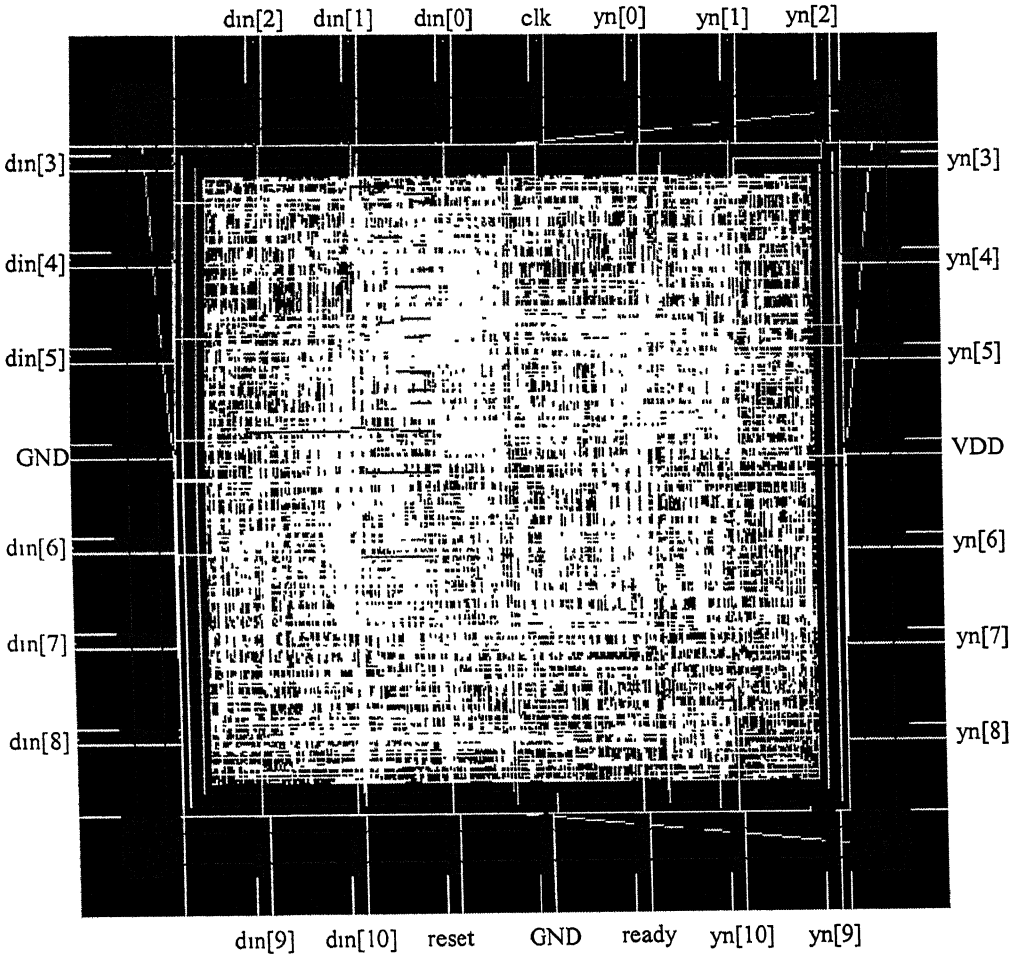

Figure 4.4. The VLSI DWT chip.

4.4 Post Layout Static Timing Analysis

Static timing analysis (STA) is a procedure in which the signal delays are calculated along each of the paths in the circuit [12]. The primary objective of post layout STA is to determine the delays along all the signal paths and verify if the actual timing

relationships between signals satisfy the desired ones. Post layout STA gives an accurate picture of the timing relationships because the signal delays along the nets due to parasitics are accurately determined after placement of cells and routing of wires. This is an improvement over the timing estimates obtained at the gate level due to the following reasons.

1. No information regarding placement of cells and routing of wires exist at the gate level of design flow. Statistical wire load models are used for estimation of delays along nets. Hence, parasitics are not taken into account accurately.
2. The clock tree information does not exist at the gate level. Therefore ideal clocks are assumed for timing analysis.

A second objective of post layout STA is to determine the minimum cycle time required for the clock signal for proper operation of the circuit. This in effect determines the maximum speed at which the circuit may work.

In the present work, the delays along all the nets were extracted in Standard Delay Format (SDF) [12], and the circuit was analyzed for timing by back annotating actual delays to their respective paths in the netlist using Pearl Static Timing Analyzer [16]. The delays along the maximum and minimum delay paths are shown in Figs. 4.5 and 4.6 respectively.

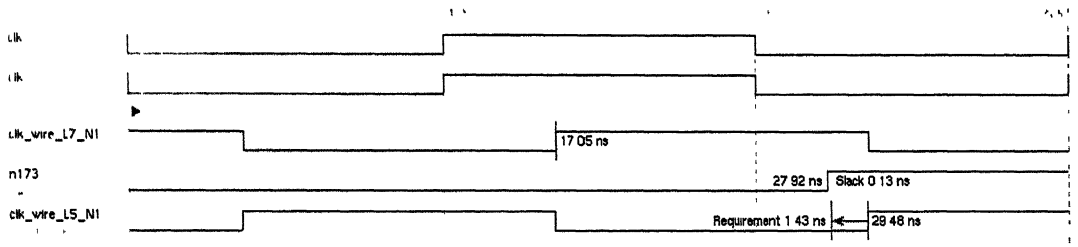

Figure 4.5. Setup time check for maximum delay path.

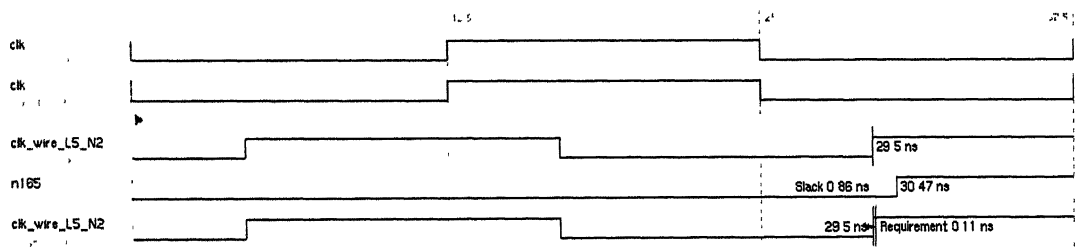

Figure 4.6. Hold time check for minimum delay path.

A clock signal of cycle time 25 ns was applied for analysis. A slack value of 0.13 ns in Fig. 4.5 implies that the signal arrives 0.13 ns before the time after which it would cause setup time violations. Similarly, a slack value of 0.86 ns for the minimum delay path in Fig. 4.6 indicates that data changes 0.86 ns after the time before which hold time violations would occur. The maximum delay path deduced from the post layout STA is shown in Fig. 4.7. The minimum cycle time required for proper operation was determined to be 24.74 ns.

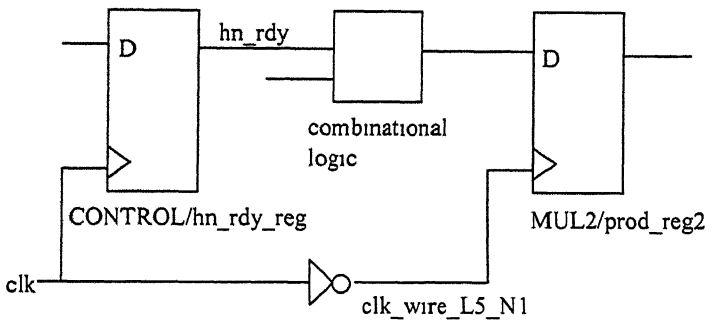

Figure 4.7 The maximum delay path at physical design level.

It can be seen that the maximum delay paths in gate level design and post layout design are different. A comparison of delays of different blocks and the minimum cycle times at gate level and physical design level is given in Table 4.4.

Table 4.4. Comparison of delay and minimum cycle time values at gate level and physical design level

Timing	Gate level estimate	Post layout value
Maximum multiplexer (MUX) delay	2.33 ns	1.68 ns
Maximum multiplier (MUL) delay	12.68 ns	12.06 ns
Maximum adder unit (ADDR) delay	7.61 ns	8.21 ns
Minimum cycle time of clock signal	22.47 ns	24.74 ns

It can be concluded from the above observations that the VLSI DWT circuit can work at a maximum frequency of about 40 MHz under TYPICAL operating conditions.

4.5 Discussion of Results

One of the important applications of the discrete wavelet transform is in video signal processing. Assuming a frame rate of 30 frames per second (NTSC standard), each

colour frame of 512 x 512 pixels of a video signal must be processed within 33 ms for real time operation. At 40 MHz clock speed, the computations on a single frame takes 19.5 ms. Thus, the circuit satisfies real time requirements for colour video processing employing the DWT. An additional circuit must be used to compute the 2-D DWT for image or video signals.

4.5.1 Review of decisions

It is necessary to review the decisions taken during the design process in light of the results obtained. The objective of such an analysis is to identify those decisions that may need to be changed in order to obtain a more optimal design or to satisfy the specifications that were not met. In the design of the DWT circuit, timing aspects were emphasized. The effect on speed and area is studied here. The area occupied by different resources is enumerated in Table 4.5.

Table 4.5. Area of different resources as part of the total core area

Description	Area occupied	Percentage of core area
One 11 bit-register	13582.1	0.34
12, 8 bit x 11 bit multipliers	2883624.0	71.76
10, 19 bit adders	253601.2	6.31
12, 19-bit pipeline registers	281519.8	7.00
Control unit	25090.5	0.624
6, 11-bit registers (STG1)	737160.5	1.834
5, 11-bit registers (STG2)	59719.6	1.486
6, 11-bit registers (INP_DEL)	95696.6	2.381
6, 11-bit 4 x 1 multiplexers	178537.2	4.443
Output multiplexer	5287.6	0.131
Output register (11 bits)	13582.1	0.34
12, 8-bit registers	120683.5	3.00

It can be seen that, the multipliers occupy the highest area. Twelve multipliers were used for design at the RTL level of design flow. No attempt was made to reduce the number of multipliers to 6 for 6-tap filters as in [7] because the speed of computation for the schedule used here would be reduced by a factor of 2. From the post layout STA results

it can be inferred that the maximum speed of operation can only be 20 MHz with the technology library used. At this speed, real time requirements for colour video signals are not satisfied. Hence other methods must be used to reduce the area of multipliers.

At the RTL level, it was also decided to pipeline the filter unit so that the critical path in terms of path delay is lesser than that of filter unit shown in Fig. 3.8. From the gate level timing analysis the maximum delay path was deduced to be the path shown in Fig 4.1 and the minimum cycle time was derived to be 22.47 ns. If this had remained a maximum delay path even after layout, the required minimum cycle time would have been 20.77 ns (i.e., a maximum speed of 48.14 MHz). The actual critical path shown in Fig. 4.7 imposed a minimum cycle time of 24.74 ns. For the filter unit as in Fig. 3.8, one of the possibilities for maximum delay path would have included the ADDR unit delay as shown in Fig. 4.8.

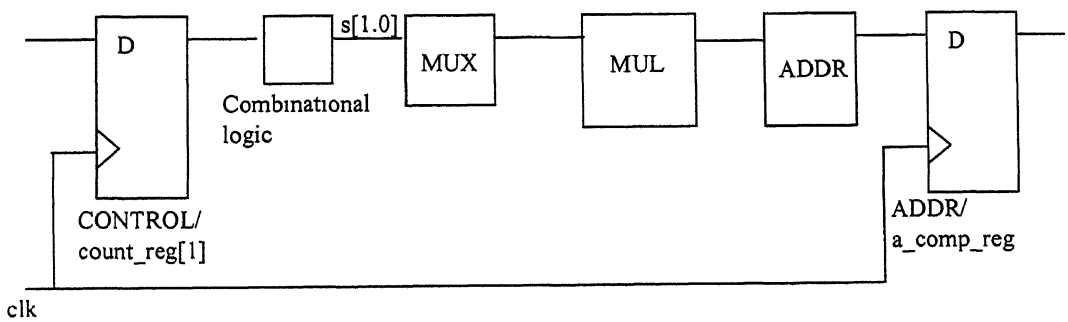

Figure 4.8. A possible maximum delay path with a non-pipelined filter.

Calculating the delays along the path based on the STA values, the minimum cycle time requirement can be determined to be 28.98 ns. Hence the maximum speed of operation would atmost be 34.5 MHz. Although, pipelining the filter unit as in Fig. 3.9 can result in an improvement in speed by about 13.6 MHz at the cost of 7% increase in area, an improvement of only 5 5 MHz was achieved due to the bottleneck imposed by the control signal delay.

4.5.2 Analysis of maximum delay path

From Fig. 4.7 it can be seen that the registers CONTROL/hn_rdy_reg and product register MUL2/prod_reg are triggered at different clock edges of the same cycle. Hence a half-cycle relationship exists between the two registers. A significant fraction (56%) of the signal ('hn_rdy') delay was determined to be due to clock-to-Q delay of

CONTROL/hn_rdy_reg register. Since a clock signal of 50% duty cycle is being used, twice the delay is actually included in deriving the minimum cycle time. This has resulted in a longer cycle time. Hence it can be concluded that the paths related to 'hn_rdy' must be separately targeted for optimization in order to remove the bottleneck.

4.5.3 Disadvantages of the circuit

The VLSI DWT circuit described here is not optimal with respect to area. This was mainly attributed to 12 multipliers being used as part of two filter units and the twelve 19 bit registers used for pipelining the filter units. Such a filter structure may not be justified unless the bottleneck due to the control signal is removed for significant improvement in speed.

In choosing the word lengths for filter and intermediate DWT coefficients, care was taken to see that overflow errors did not occur. However, certain applications in the domain of signal processing require specific signal-to-noise ratios (SNR) to be met. This in effect determines the acceptable maximum errors due to truncation and quantization. Truncation and quantization errors were not considered in the determination of word lengths for this implementation.

Chapter 5

Conclusion and Scope for Future Work

5.1 Conclusion

The design of a VLSI discrete wavelet transform circuit starting from a behavioral model to physical design, passing through the different levels of VLSI design flow was described in this thesis. The circuit is capable of computing the DWT of a sequence of input samples in a continuous fashion using wavelet filters with upto 6 coefficients and for 3 levels of decomposition. The computational resources include two filter units consisting of 6 multipliers and 5 adders each. The filter units were pipelined by 2 stages. The latency of the filter units is 2 clock cycles while the throughput of the circuit is one DWT coefficient output per clock cycle. The design was optimized with respect to speed, first at the RTL level by choosing the computation schedule of Table 2.2 to facilitate pipelining of filters and hence at gate level in form of applying timing constraints during logic synthesis. Constraints were also applied at the physical design level to synthesize the clock tree with minimum skew.

It was determined from the static timing analysis of the circuit that it can function at a maximum clock frequency of 40 MHz under typical conditions. At this speed colour video signals can be processed with DWT in real time.

From a discussion of results and analysis of maximum paths, it was apparent that, while attempts were made to reduce only the data path delay, the delay along a control signal proved to be a bottleneck. Upto 39% improvement in speed can be obtained if the filter unit in Fig. 3.9 is used for realizing the DWT circuit at the cost of 7% increase in area if the bottleneck due to the control signal is removed.

5.2 Extensions to this work

In this work, attempts were made to optimize the circuit with respect to only one parameter of the design space, namely, speed. Most often, area and power also constitute important parameters of the design space. It is therefore desirable to realize a VLSI DWT circuit that is optimal with respect to all the three parameters. The design of such circuits is however, complex. The decisions taken to optimize the circuit with respect to one of the parameters may have an impact on other parameters as was apparent in this implementation. In this work, optimizations with respect to speed led to increase in the

area of the system. Such a design process requires much iteration to arrive at the most optimal solution. The design of a VLSI DWT circuit that is optimal with respect to all the three parameters is suggested for future work.

References

1. S. G. Mallat, *A Wavelet Tour of Signal Processing.* Chestnut Hill, MA: Academic Press, 1998.

2. O. Rioul, M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine,* pp. 14-38, October 1991.

3. N. Weste, K. Eshragian, *Principles of CMOS VLSI Design.* Reading, MA: Addison-Wesley, October 1999.

4. P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

5. G. Knowles, "VLSI architecture for the discrete wavelet transform," *Electronics Letters,* vol. 26, no. 15, pp. 1184-1185, July 1990.

6. K. K. Parhi, T. Nishitani, "VLSI architectures for discrete wavelet transforms," *IEEE Transactions on VLSI Systems,* vol. 1, no. 3, pp. 191-202, June 1993.

7. A. Grzeszczak, M. K. Mandal, S. Panchanathan, T. Yeap, "VLSI implementation of the discrete wavelet transform," *IEEE Transactions on VLSI Systems,* vol. 4, no. 4, pp 421-433, December 1996

8. S. G. Mallat, "A theory for multiresolution signal decomposition: A wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 674-693, 1989.

9. M. Vishwanath, "The recursive pyramid algorithm for discrete wavelet transform," *IEEE Transactions on Signal Processing,* vol. 48, no. 3, pp. 673-676, March 1994.

10. S. Sjoholm, L. Lindh, *VHDL for Designers.* Hertfordshire: Prentice-Hall Europe, 1997.

11. Y. F. Huang, C. H. Wei, *Circuits and Systems in the Information Age.* IEEE Press, 1997.

12 H. Bhatnagar, *Advanced ASIC Chip Synthesis.* Kluwer Academic Publishers, 1999.

13. Synopsys Inc., *Design Compiler Reference Manual: Constraints and Timing.* Version 1999.10, October 1999.

14. I. S. Kourtev, E. G. Friedman, *Timing Optimization through Clock Skew Scheduling.* Norwell, MA: Kluwer Academic Publishers, 2000.

15. Cadence Design Systems, *Envisia Silicon Ensemble Place and Route Manual.* December 1998.

16. Cadence Design Systems, *Pearl Static Timing Analyzer Manual*. December 1998.

17. J Cavanagh, *Digital Computer Arithmetic*. New York: McGraw-Hill, 1984.

18. J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*. New Delhi: Prentice-Hall of India, 2000.

A 133750

Date Slip

The book is to be returned on the date last stamped.

.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....